PY1003

Lecture 7: Trees for Sentential Logic IV

● A set of sentences is inconsistent iff the conjunction of all the sentences in the set is a contradiction. E.g. {A, ¬A} is inconsistent and A∧¬A is a contradiction; {A, B, C} is consistent and (A∧B)∧C is not a contradiction.

More on Exhibiting Models

Starting sentence: ¬(A∧B)∧(A→B)

```
     ¬(A∧B)∧(A→B)
         ¬(A∧B)
         (A→B)
         /     \
      ¬A        ¬B
     /  \      /  \
   ¬A    B   ¬A    B
```

The tree is finished and has open branches, so this sentence is consistent. The rightmost open branch tells us that one model is:
I(A)=F
I(B)=F

The middle open branch tells us that another is:
I(A)=F
I(B)=T

The leftmost open branch tells us that any interpretation which makes ¬A true (i.e. makes A false) will be a model – it doesn't matter what value B takes. But we have already listed both the interpretations which make A false as models.

Compare the truth table:

	A	B	¬(A∧B) ∧ A→B			
1	T	T	f	t	F	t
2	T	F	t	f	F	f
3	F	T	t	f	T	t
4	F	F	t	f	T	t

Lines 3 and 4 represent the same two models for our starting sentence as we identified using the tree method.

Testing the validity of Sentential arguments with truth trees

● An argument is valid iff it is impossible for the premises to be true and the conclusion false
i.e. iff there is no interpretation which makes the premises true and the conclusion false
i.e. iff there is no interpretation which makes the premises true and the negation of the conclusion true.

● This means we can test whether an argument is valid by testing whether there is an interpretation which makes the premises true and the negation of the conclusion true
i.e. by testing whether the set consisting of the premises and the negation of the conclusion is a consistent set. This is something we already know how to test.

● For instance, we can test whether the argument
A→B, AvB ├ B
is valid by testing whether the set
{A→B, AvB, ¬B}
is consistent. If the set is inconsistent, the argument is valid: there is no way for the premises to be true and the conclusion false. If the set is consistent, the argument is invalid: there is a way for the premises to be true and the conclusion false.

Example 1

Is the following argument valid?
A→B, AvB ├ B
To answer this, we test whether {A→B, AvB, ¬B} is consistent.

```
      A→B
      AvB
       ¬B
      /   \
    ¬A     B
    /  \
   A    B
```

Every branch is closed, so the set comprising the premises plus the negated conclusion is inconsistent. Therefore the argument we started with is valid.

Example 2

Is the following argument valid?
A, A↔(BvC) ├ C
To answer this, we test whether {A, A↔(BvC), ¬C} is consistent.

```
          A
      A↔(BvC)
         ¬C
        /    \
      A      ¬A
  (BvC)      ¬(BvC)
   /   \
  B     C
```

The tree is finished and has an open branch, so the set is consistent. Therefore the argument we started with is invalid.

Exhibiting a counterexample to an argument

A counterexample to an argument is an interpretation which makes all the premises true and the conclusion false. A counterexample exists iff the argument is invalid. **A counterexample for an argument is a model for the set consisting of all the premises and the negated conclusion.**

The open branch on the tree above can be used to construct a model for the set of sentences {A, A↔(BvC), ¬C}:
I(A) = T
I(B) = T
I(C) = F
This interpretation is a counterexample to the argument A, A↔(BvC) ├ C. On this interpretation, all the premises are true and the conclusion is false.

Arguments from no premises

An argument is valid iff it is impossible for the premises to be true and the conclusion false. So an argument automatically counts as valid when it is impossible for the premises to be true (i.e. if the premises are jointly inconsistent). An argument also automatically counts as valid when it is impossible for the conclusion to be false (i.e. if the conclusion is a tautology).

Sometimes we want to say that a sentence which cannot be false 'follows from no premises'. What that means is that an argument with this sentence as its conclusion would be valid *whatever premises it had* – even none at all. We express an argument from no premises by writing its conclusion after a turnstile, with nothing before the turnstile:

├ A→(AvB)

We can test this argument for validity, like any other, by testing whether it is impossible for its premises to be true and its conclusion false. Since there are no premises, however, this amounts to testing whether it is impossible for the conclusion to be false, i.e. whether the conclusion is a tautology.

The argument $\vdash A \to (A \vee B)$ is valid just in case the sentence $A \to (A \vee B)$ is a tautology.

$\neg(A \to (A \vee B))$
|
A
$\neg(A \vee B)$
|
$\neg A$
$\underline{\neg B}$

Summary of the procedure for testing for validity:

1. List the premises (if any), followed by the negated conclusion.
2. Complete a truth tree for these sentences.
3. If all the branches close, the argument is valid.
4. If any branch remains open, the argument is invalid.
5. Any open branches can be used to find counterexamples to the argument.

Examples

Are the following arguments valid? Where an argument is invalid, exhibit a counterexample.

1. $A \wedge (B \wedge C) \vdash A \vee (B \vee C)$

2. $\vdash (A \wedge (A \to B)) \to B$

3. $C \leftrightarrow D \vdash C \vee D$

4. $A \wedge (\neg B \to A) \vdash \neg(A \to B)$

Before we move on to Predicate Logic in the next lecture, you should make sure you are confident that you can use Sentential truth trees to:

1. Test whether a sentence, or set of sentences, is consistent
2. Exhibit a model for a consistent sentence or set of sentences
3. Test whether a sentence is a contradiction
4. Test whether a sentence is a tautology
5. Test whether an argument is valid (including arguments from no premises)
6. Exhibit a counterexample for an invalid argument